

实验 1 操作环境与过程

1.1 实验目的

1. 学习如何在 VC++ 6.0 环境中编写 C 语言程序。
2. 学习如何在 VC++ 6.0 环境中编译、连接和运行 C 语言程序。

1.2 实验预习

1. 熟悉文字的编辑修改。
2. 熟悉 C 语言程序的基本结构。
3. 熟悉 C 语言注释的表示方法。
4. 理解 C 语言原程序转换为可执行代码的过程。

1.3 使用 VC++ 6.0 上机过程

【步骤 1】选择[开始]→[程序]→[Microsoft Visual Studio]→[Microsoft Visual C++]菜单选项进入 VC++ 6.0，出现如图 1.1 所示界面。

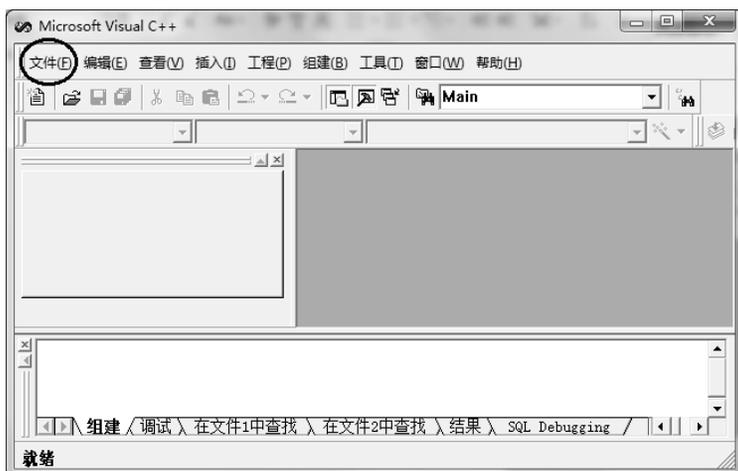


图 1.1 VC++ 6.0 主界面

【步骤 2】选择[文件]→[新建]菜单选项，打开“新建”对话框，在[文件]选项卡中，选择[C++ Source File]选项，如图 1.2 所示。

【步骤 3】选择保存文件的位置，并输入文件名（如图 1.2 所示），扩展名可以省略，系统将自动添加扩展名.cpp，单击[确定]按钮，在打开的界面中开始输入源程序，如图 1.3 所示。



图 1.2 “新建”对话框

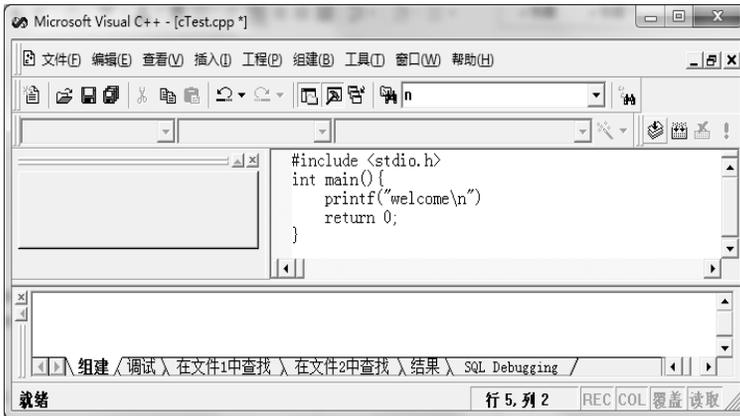


图 1.3 输入源程序

【步骤 4】选择[组建]→[编译]菜单选项，如图 1.4 所示，编译源程序，第一次编译会出现建立工作区请求，单击[是]按钮，如图 1.5 所示。

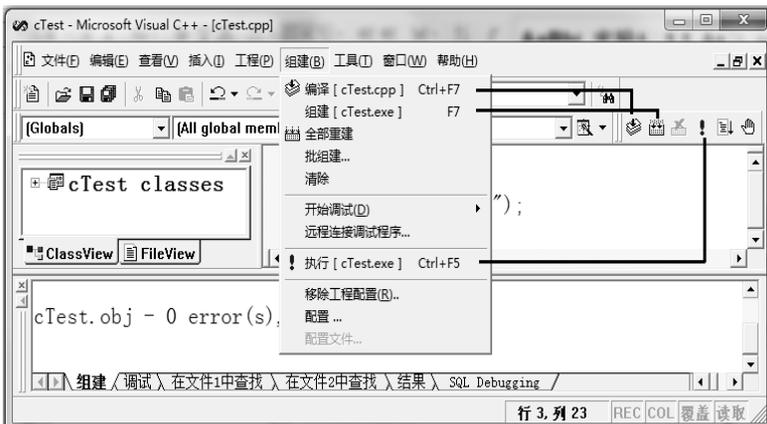


图 1.4 [编译]菜单选项

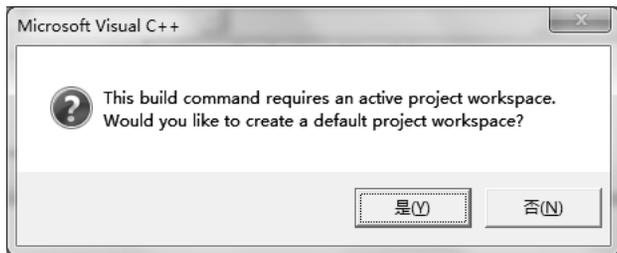


图 1.5 建立工作区请求

在出现保存请求时单击[是]按钮，如图 1.6 所示。



图 1.6 保存请求

【步骤 5】查看调试窗口内有无提示错误信息，若有，则根据提示的行号和信息进行修改，例如：若源代码中语句缺少结束符号“;”，则会出现如图 1.7 所示的错误提示信息。

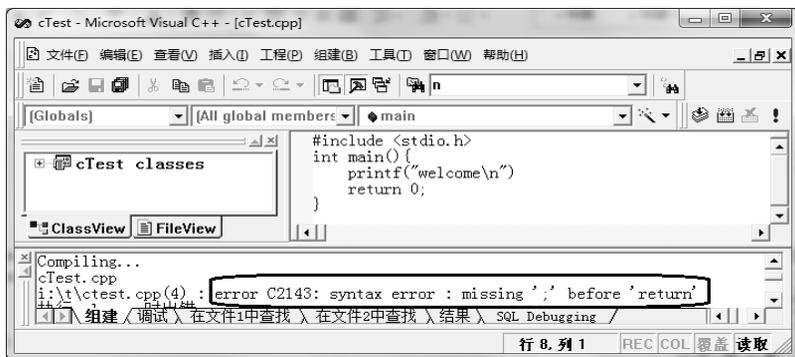


图 1.7 语句缺少“;”错误提示

【步骤 6】改正后错误重新编译，直至出现 0 个错误提示后，选择[组建]→[组建]菜单选项进行链接，若有错则改，若没有错误就可以选择[组建]→[执行]菜单命令开始运行程序了，然后在控制台窗口内查看结果，如果结果不对，则修改程序，重复上面的过程，如图 1.8 所示。



图 1.8 查看结果

【步骤 7】在编译链接过程中也可以使用工具栏按钮，也可以不选择编译，而直接选择运行，系统会自动先编译、链接后执行，如图 1.9 所示。



图 1.9 工具栏按钮

【步骤 8】选择[关闭工作空间]菜单选项关闭当前文档窗口和工作空间，开始下一个程序，如图 1.10 所示。

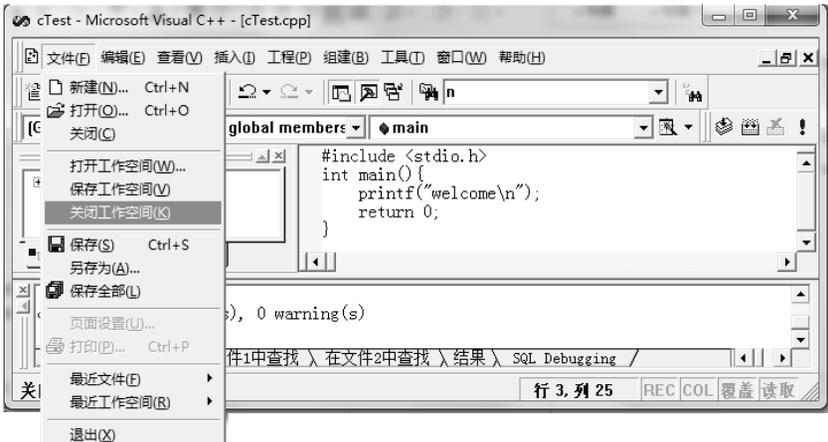


图 1.10

1.4 使用 Dev-C++ 上机过程

【步骤 1】执行 Dev-C++ 后会出现如图 1.11 所示界面。

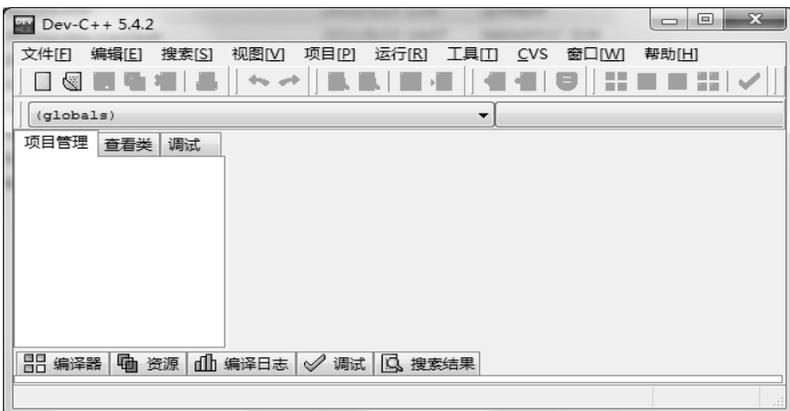


图 1.11 Dev-C++ 界面

【步骤 2】选择[文件]→[新建]→[源代码]菜单选项，输入源代码（图 1.12）。

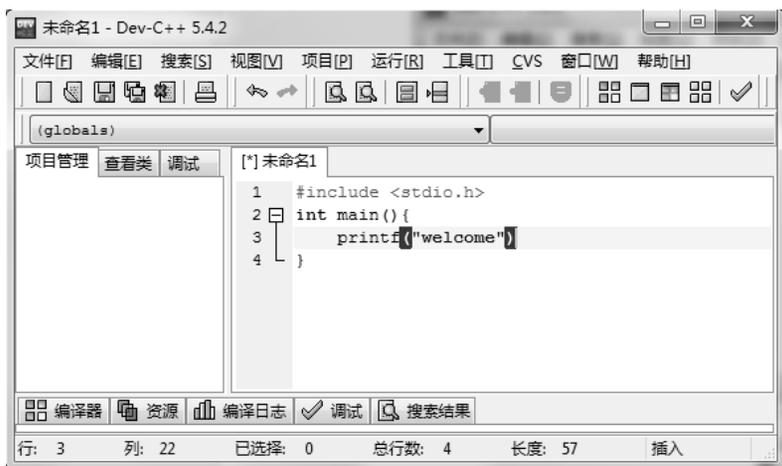


图 1.12 输入源代码

【步骤 3】选择[运行]→[编译]菜单选项或单击对应按钮，如图 1.13 所示。

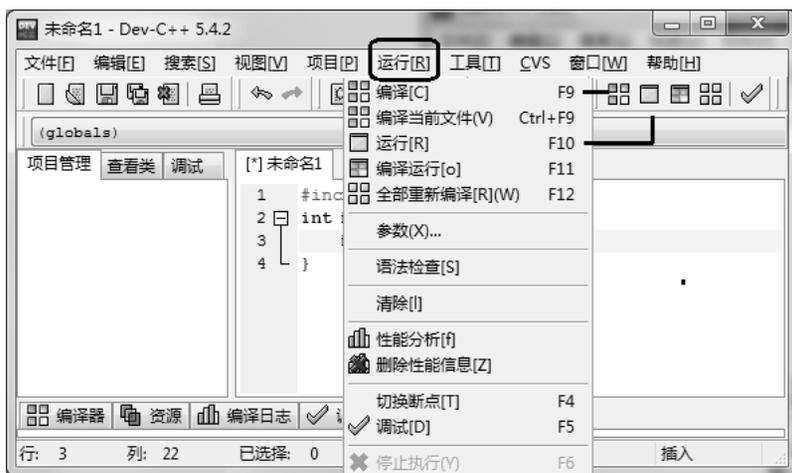


图 1.13 [编译]菜单命令

【步骤 4】在弹出的询问对话框中单击[Yes]按钮，如图 1.14 所示，然后选择保存的位置和文件名，如图 1.15 所示。



图 1.14 询问对话框



图 1.15 “保存文件”对话框

【步骤 5】若发现错误，如图 1.16 所示，则修改错误后，重新编译。

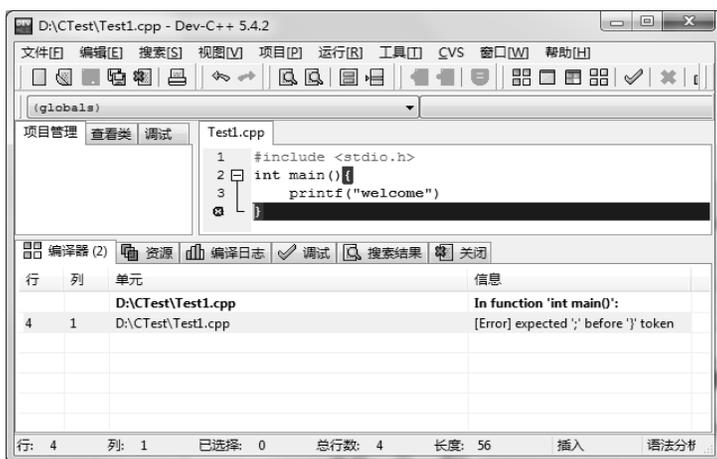


图 1.16 发现错误

【步骤 6】直到没有错误后，选择[运行]菜单命令或单击相应按钮，如图 1.13 所示，查看运行结果，如图 1.17 所示。

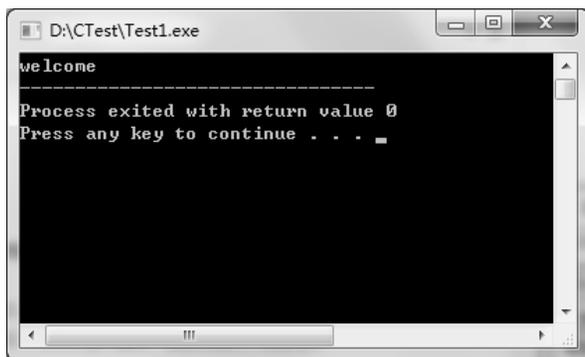


图 1.17 查看运行结果

【步骤 7】一个程序调试完成后，选择[文件]→[全部关闭]菜单命令，然后重新开始。

1.5 在 VS 2013 中如何调试 C 语言程序

【步骤 1】选择[文件]→[新建]→[项目]菜单命令，如图 1.18 所示。



图 1.18 [项目]菜单命令

【步骤 2】选择[Visual C++]→[Win32]→[Win32 控制台应用程序]，设置项目名和存储位置后，单击[确定]按钮完成项目的新建，如图 1.19 所示。

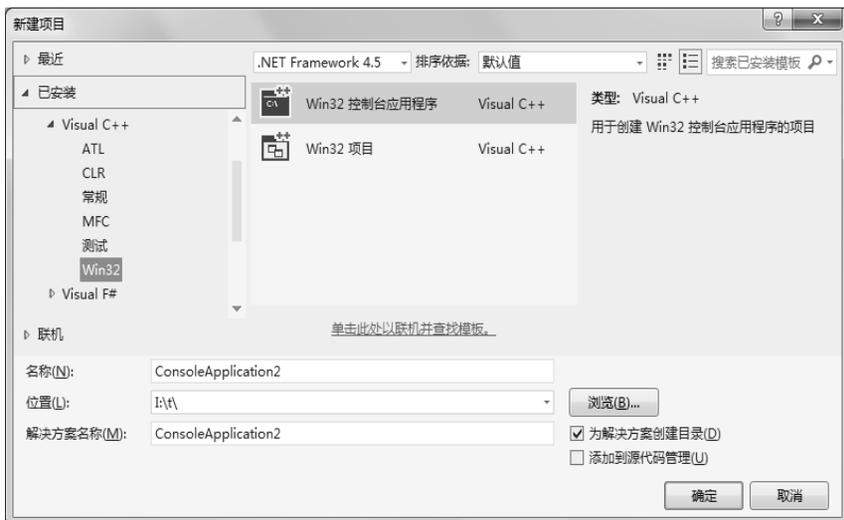


图 1.19 新建项目

【步骤 3】原有代码不必改动，添加程序代码，如图 1.20 所示。

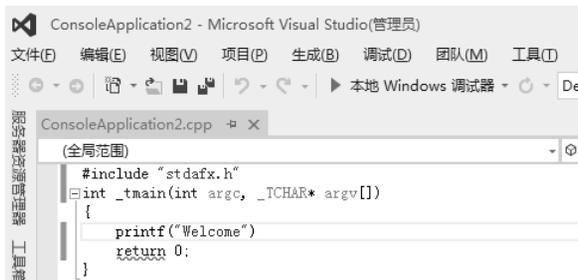


图 1.20 添加程序代码

【步骤 4】单击[生成]→[生成解决方案]菜单命令，根据错误提示修改错误，如图 1.21 所示，然后重新生成，直到成功。



图 1.21 根据错误提示修改错误

【步骤 5】单击[调试]→[开始执行]菜单命令，结果如图 1.22 所示。



图 1.22 执行程序

【步骤 6】关于自动生成源代码的说明。

stdafx.h 是必须保留的，其内容如下：

```
#include "targetver.h"
#include <stdio.h>
#include <tchar.h>
```

可以看出，其中已包含了 stdio.h，因此 stdio.h 就不需要再写了，需要其他函数时，仍然需要添加，stdafx.h 中还增加了对 tchar 即 unicode 的支持。

入口函数改为 int _tmain(int argc, _TCHAR* argv[])，代替了原来的 int main()函数。

1.6 编译时常见错误

写完程序后，先要进行编译，大量的语法错误在此时便会出现，常见的语法错误有以下几种。

1. Cannot open include file

嵌入文件打不开，`#include` 后的文件名拼写错误。

2. syntax error : missing ';'

缺少“;”漏写“;”或有多余的“}”。

3. newline in constant

常量跨行漏写“”。

4. unexpected end of file found

未发现文件结尾，缺少“}”或“{”不配套。

5. unknown character

错用了中文标点。

6. XXX:undeclared identifier

不认识的关键字，一般是变量名未声明，或未加函数所在的头文件。

1.7 链接时常见错误

一个程序经过编译没有错误，仅仅表示没有语法错误了，但是链接时还可能会出现错误，链接时的错误主要有以下三种。

1. unresolved external symbol `_main`

无法链接上 `main` 函数（如写成了 `mian`）。

2. `_main` already defined in XXX

`main` 函数已经存在，一个程序中出现多个 `main` 函数。通常出现这种情况的原因是，写完程序后，没有关闭工作空间，又建立了新文件，导致一个程序，两个文件，都有 `main` 函数。解决办法是打开工作空间的文件页签，删除一个不需要的文件，如图 1.23 所示。

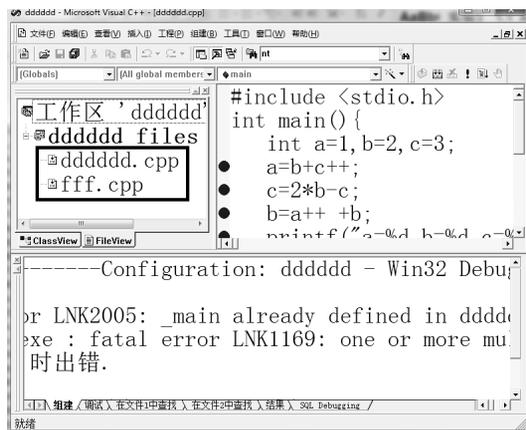


图 1.23 删除一个不需要的文件

删除方法是单击选中要删除的文件，按 Delete 键。

3. cannot open debug/xxx.exe for writing

无法写入 debug/xxx.exe，原因是上次执行的 Console 窗口未关闭，可以直接关闭。如果关闭比较困难，可以选择把 VC++ 关闭后再重新打开，也可以按 Ctrl+Alt+Del 组合键，在“Windows 任务管理器”窗口中，找到运行的进程，强行关闭，如图 1.24 所示。

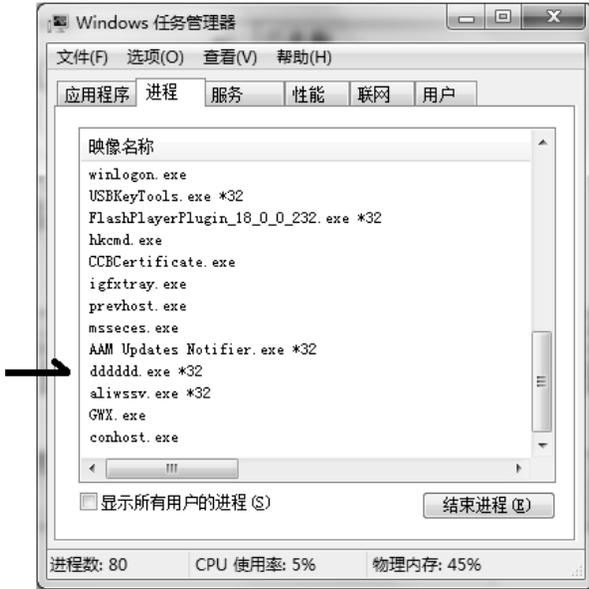


图 1.24 “Windows 任务管理器”窗口

1.8 常见错误及注意事项

(1) 双击错误提示信息，可以使光标出现在错误代码所在的行，加快错误定位。

(2) 缺少“;”错误往往定位显示在下一行，因为 C 语言语句是可以跨行的。

(3) 注意一个程序调试完成后，一定要调用[文件]→[关闭工作空间]关闭工作区，否则在一个项目内会出现两个 main 函数，无法通过链接操作。